IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Before the Board of Patent Appeals and Interferences

App. No.	:	10/089,139	Confirmation No.:	2275
Inventor	:	Bosworth et al.		
Filed	:	August 19, 2002		
Title	:	A MULTI-LANGUAGE EXECUTION METHOD		
Art Unit	:	2191		
Examiner	:	Rampuria, Satish		
Customer No.	:	25,943		

MAIL STOP: APPEAL BRIEF-PATENTS
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

APPELLANT'S BRIEF IN SUPPORT OF APPELLANT'S APPEAL TO THE BOARD OF PATENT APPEALS AND INTERFERENCES

Dear Sir:

In response to the Notification of Non-Compliance mailed, Appellant hereby re-submits the Appeal Brief previously submitted on August 13, 2007, in support of Appellant's appeal. This is a re-submission of Appellant's Brief in response to the Notification of Non-Compliance mailed on November 16, 2007. The deficiency has been corrected. This appeal furthers the Notice of Appeal filed on June 13, 2007. The appeal arises from a final decision by the Examiner in the final Office Action, dated March 22, 2007. The final decision was in response to arguments filed on December 27, 2006, in response to an earlier office action, mailed October 4, 2006.

Appellant re-submits this *Brief on Appeal*. Payment in the amount of $500.00 to cover the fee for filing the *Brief on Appeal* was tendered with the original submission.

Appellants respectfully request consideration of this appeal by the Board of Patent Appeals and Interferences for allowance of the present patent application.

Real Party in Interest:

The Real Party in Interest is BEA Systems, Inc., of 2315 North First Street, San Jose, California 95131, assignee of the application. Assignment of the application from the Inventors to BEA Systems, Inc. is recorded with the United States Patent and Trademark Office on August 19, 2002, at Reel 013198 Frame 0130.

Related Appeals and Interferences:

To the best of Appellants' knowledge, there are no related appeals or interference proceedings currently pending, which would directly affect or be directly affected by or have a bearing on the Board's decision in this appeal.

Status of Claims:

Appellants appeal the rejection of claims 1-13 and 20-32, which were rejected in the final Office Action dated March 22, 2007. Claims 1-13 and 20-32 are reproduced, as pending, in Appendix A. Claims 14-19 and 33-38 were previously cancelled and are no longer pending.

Status of Amendments:

Appellants have cancelled claims 14-19 and 33-38, without prejudice in Appellants' Response to Final Office Action dated April 25, 2007. The cancellations were offered to reduce the issues on Appeal in conformance with 37 CFR 1.116, notwithstanding Appellants disagreement with the Examiner's ground for rejection. In the Examiner's subsequent Advisory Action, mailed May 14, 2007, the Examiner accepted the after-final cancellations of claims 14-19 and 33-38.

Summary of the Claimed Subject Matter:

Independent claim 1 is as follows. Support for each limitation of claim 1 in the form of figure elements corresponding to each limitation and portions of the Specification given by page and line numbers for each limitation is shown, inline:

"1. A method of computing comprising:" **(Figures 2a-2c; Page 9, line 14 through page 11, line 23.)**

"reading, by an execution engine, a data processing representation having code sections with code statements of at least a first and a second programming language;" **(Elements 202 of Figure 2a; Page 9, lines 22-25.)**

"recognizing, by the execution engine, a first code section with at least code statements of a first programming language;" **(Element 204 of Figure 2a; Page 9, lines 25-33.)**

"invoking, by the execution engine, a first code statement processing unit of the first programming language to process the first code section;" **(Element 204 of Figure 2a; Page 9, lines 25-33.)**

"recognizing, by the execution engine, a second code section with at least code statements of a second programming language;" **(Element 204 of Figure 2a; Page 9, lines 25-33.)**

"invoking, by the execution engine, a second code statement processing unit of the second programming language to process the second code section." **(Element 204 of Figure 2a; Page 9, lines 25-33.)**

Independent claim 20 is as follows. Support for each limitation of claim 20 in the form of figure elements corresponding to each limitation and portions of the Specification given by page and line numbers for each limitation is shown, inline:

"20. An apparatus comprising:" **(Element 102 of Figure 1; Page 4, line 25 through page 6, line 11. Element 300 of Figure 3; Page 11, line 25 through page 12, line 11.)**

"at least one storage unit having stored thereon programming instructions designed to instantiate an execution engine to enable the apparatus to"

(Elements 304, 306, 314a, and 314b of Figure 3; Page 11, line 25 through page 12, line 11.)

"read, by the execution engine, a data processing representation having code sections with code statements of at least a first and a second programming language," **(Elements 202 of Figure 2a; Page 9, lines 22-25.)**

"recognize, by the execution engine, a first code section with code statements of at least the first programming language," **(Element 204 of Figure 2a; Page 9, lines 25-33.)**

"invoking, by the execution engine, a first code statement processing unit of the first programming language to process the first code section," **(Element 204 of Figure 2a; Page 9, lines 25-33.)**

"recognize, by the execution engine, a second code section with code statements of at least the second programming language," **(Element 204 of Figure 2a; Page 9, lines 25-33.)**

"invoking, by the execution engine, a second code statement processing unit of the second programming language to process the second code section; and" **(Element 204 of Figure 2a; Page 9, lines 25-33.)**

"at least one processor coupled to said at least one storage unit to execute said programming instructions." **(Elements 302-306, 314a, and 314b of Figure 3; Page 11, line 25 through page 12, line 11.)**

Grounds for Rejection to be Argued on Appeal:

I. Claims 20-32 stand rejected under 35 U.S.C. §112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.

II. Claims 1-3, 6-7, 20-22, and 25-26 stand rejected under 35 U.S.C. §102(e) as being anticipated by U.S. Patent No. 6,292,936 to *Wang* (hereinafter "Wang").

III. Claims 4, 5, 8, 23, 24 and 27 stand rejected under 35 U.S.C. §103(a) as being unpatentable over Wang in view of U.S. Patent No. 6,732,330 to *Claussen* (hereinafter "Claussen").

IV. Claims 9-13 and 28-32 stand rejected under 35 U.S.C. §103(a) as being unpatentable over Wang in view of U.S. Patent No. 5,428,792 to *Conner et al.* (hereinafter "Conner").

Arguments:

 I. Rejection of claims 20-32 under 35 U.S.C. §112, second paragraph was improper because claims 20-32 particularly point out and distinctly claim the subject matter which applicant regards as the invention.

  Appellants respectfully disagree with the Examiner's rejection. No specific reasons are provided for the rejection of claims 20-32 under §112, second paragraph. The only specific reason given is only mentioned in regard to claims 14-19 and 33-38, which were cancelled in Appellant's Response to the Final Office Action, mailed April 25, 2007. Further, the reasons given for rejecting claims 14-19 and 33-38 do not apply to claims 20-32. Claims 14-19 and 33-38 were rejected for reciting "adapted to" and "first and second code statements." Claims 20-32 do not include these recitations.

  Accordingly, without knowing any sort of specific reason, and upon Appellants' own further review of the claims, Appellants maintain that claims 20-32 are not indefinite under §112, second paragraph.

II.   Rejection of claims 1-3, 6-7, 20-22, and 25-26 under 35 U.S.C. §102(e)
        was improper because Wang fails to teach each and every limitation.

It is well settled that anticipation under 35 U.S.C. §102 requires the disclosure in
a signal piece of prior art to teach **each and every** limitation of a claimed invention.
Electro Med. Sys. S.A. v. Cooper Life Sciences, 34 F.3d 1048, 1052, 32 USPQ2d 1017,
1019 (Fed. Cir. 1994). . MPEP 2131 states, "TO ANTICIPATE A CLAIM, THE
REFERENCE MUST TEACH EVERY ELEMENT OF THE CLAIM" and "a claim is
anticipated only if each and every element as set forth in the claim is found, either
expressly or inherently described, in a single prior art reference." *Verdegaal Bros. v.
Union Oil Co. of California*, 814 F.2d 628, 631, 2 USPQ2d 1051, 1053 (Fed. Cir. 1987).

Furthermore, anticipation requires that each claim element must be <u>identical</u> to a
corresponding element in the applied reference. *Glaverbel Société Anonyme v.
Northlake Mktg & Supply, Inc.*, 45 F.3d 1550, 1554 (Fed. Cir. 1995). Thus, to anticipate
the present invention, Wang must disclose every element recited in the pending claims.

Claim 1 calls for a "method of computing comprising:
reading, by an execution engine, a data processing representation having code
        sections with code statements of at least a first and a second
        programming language;
recognizing, by the execution engine, a first code section with at least code
        statements of a first programming language;
invoking, by the execution engine, a first code statement processing unit of the
        first programming language to process the first code section;
recognizing, by the execution engine, a second code section with at least code
        statements of a second programming language; and
invoking, by the execution engine, a second code statement processing unit of
        the second programming language to process the second code section."

- 6 -

In contrast, Wang fails to disclose, expressly or inherently, an execution engine that invokes first **and** second code statement processing units of first and second programming languages, as is claimed in amended claim 1. Wang merely teaches "an interpreter-based scripting environment [that] includes multiple runtime processors executed by the computer. Each of the runtime processors processes their respective corresponding intermediate sources derived from an original source in a synchronous manner" (abstract). The processors are interdependently invoked. Specifically, the original source disclosed in Wang comprises an HTML document with embedded Visual Basic scripting langauge blocks. Wang teaches a <u>single</u> HTML parser that parses the original HTML + VB source, and translates the non-VB source into a first intermediate source executable by a Java VM, and the VB source into a second intermediate source having the VB script statements executable by a VB script interpreter.

Even if we were to read the Java VM and VB Script Interpreter as the recited first and second code statement processing units, Wang does not disclose an execution engine that invokes **both** of the Java VM **and** VB Script Interpreter. The HTML parser of Wang, described above, simply creates intermediate sources and <u>does not</u> invoke either of the Java VM or the VB Script Interpreter. According to Wang, col. 3, lines 57-67 and col. 4, lines 1-8, <u>the Java VM is invoked first at runtime, and the VB Script Interpreter is later invoked by the Java VM</u>. Thus, there is no common execution engine that invokes both the Java VM and the VB Script Interpreter and, therefore, Wang does not disclose the execution engine recited by claim 1.

Both Wang and the invention of claim 1 certainly teach methods of processing multi-language specifications. But Wang teaches an alternative solution to that proposed by the claimed invention of claim 1. In Wang, no common execution engine controls the "hand off" of execution between the code processing units. Thus, Wang inserts synchronizer tokens into the intermediate code to eliminate the need for a common execution engine. In contrast, the claimed invention of claim 1 does teach

- 7 -

such a common "execution engine" that controls the invoking of both code processing units. Thus, in the claimed invention of claim 1, no synchronizer tokens are needed. Accordingly, for the reasons given above, Wang does not anticipate claim 1 and, because Wang proposes an alternative solution teaching away from that of claim 1, Wang does not even suggest claim 1.

In the Advisory Action mailed May 14, 2007, the Examiner asserts that Applicants' argument relies on features which are not recited in the rejected claims. More specifically, the Examiner states that Applicants' argument relies on a "common process", and that no common process is recited in the claims. As can be seen above, Applicants' argument relies on a "common execution engine" that invokes the first and second code statement processing units. While "common" is not explicitly recited in the rejected claims, "execution engine" is, and that execution engine is inherently a "common execution engine." In interpreting the language of claim 1 above in light of the antecedent basis rule, the Examiner is required to treat "the execution engine", which is recited as invoking both the first and second processing units, as the same execution engine. This is what Applicants mean when Applicants argue that claim 1 teaches that the first and second code statement processing units are invoked by a common execution engine. Accordingly, Applicants' above arguments stand unrefuted.

Accordingly, amended claim 1 is patentable over Wang under §102(e).

Claim 20 is claim 1 in apparatus form, and thus recites similar limitations. Accordingly, claim 20 is also patentable over Wang for at least the above reasons.

Claims 2-3, 6-7, 22, and 25-26 and depend from claims 1 and 20, respectively, incorporating their limitations. Thus, for at least the same reasons, claims 2-3, 6-7, 22, and 25-26 are patentable over Wang.

Additionally, Wang fails to clearly and expressly disclose the limitations of claim 6. Claim 6 recites recognizing a third code section of a third language and invoking a third code statement processing unit of the third language to process the third section. Wang, in contrast, simply teaches two runtime processors (110 and 112) invoked to process intermediate sources derived from HTML and VB Script. No disclosure is made of a third source of a third language or a third runtime processor. Accordingly, for at least this additional reason, Wang fails to anticipate claim 6. Claims 7 and 8 depend from claim 6 and accordingly are also patentable for at least this additional reason.

III. Rejection of claims 4, 5, 8, 23, 24 and 27 under 35 U.S.C. §103(a) was improper because Wang and Claussen, alone or in combination, fail to teach or suggest the claimed invention when the invention as claimed in claims 4, 5, 8, 23, 24 and 27 is viewed as a whole.

Claussen fails to cure the above discussed deficiencies of Wang. Therefore, claims 1 and 20 remain patentable over Wang even when combined with Claussen.

Claims 4, 5, 8, 23, 24, and 27 depend from claims 1 and 20, respectively, incorporating their limitations. Thus, for at least the same reasons, claims 4, 5, 8, 23, 24, and 27 are patentable over Wang and Claussen, alone or in combination.

IV. Rejection of claims 9-13 and 28-32 under 35 U.S.C. §103(a) was improper because Wang and Conner, alone or in combination, fail to teach or suggest the claimed invention when the invention as claimed in claims 9-13 and 28-32 is viewed as a whole.

Conner fails to cure the above discussed deficiencies of Wang. Therefore, claims 1 and 20 remain patentable over Wang even when combined with Conner.

- 9 -

Claims 9-13 and 28-32 depend from claims 1 and 20, incorporating their limitations respectively. Thus, for at least the same reasons, claims 9-13 and 28-32 are patentable over Wang and Conner, alone or in combination.

Conclusion

Appellants respectfully submit that all the appealed claims in this application are patentable and request that the Board of Patent Appeals and Interferences overrule the Examiner and direct allowance of the rejected claims.

The fees associated with the appeal brief were submitted with the original appeal brief. We do not believe any additional fees, in particular extension of time fees, are needed. However, should that be necessary, please charge our deposit account 500393. In addition, please charge any shortages and credit any overages to Deposit Account No. 500393.

Respectfully submitted,

Date: December 4, 2007

 /Robert C. Peck/
Robert C. Peck, Reg. No. 56,826
Attorney for Appellant Applicant

Schwabe Williamson & Wyatt, P.C.
1420 Fifth, Suite 3010
Seattle, WA 98101
Tel: (206) 622-1711
Fax: (206) 292-0460

PDX/109870/130097/RPE/2291516.1

Appendix A – Appealed Claims

1. (Previously Presented) A method of computing comprising:

    reading, by an execution engine, a data processing representation having code
    sections with code statements of at least a first and a second
    programming language;

    recognizing, by the execution engine, a first code section with at least code
    statements of a first programming language;

    invoking, by the execution engine, a first code statement processing unit of the
    first programming language to process the first code section;

    recognizing, by the execution engine, a second code section with at least code
    statements of a second programming language;

    invoking, by the execution engine, a second code statement processing unit of
    the second programming language to process the second code section.

2. (Previously Presented) The method of claim 1, wherein the first and second
code sections are non-interleaved code sections.

3. (Original) The method of claim 1, wherein said second code section is
embedded within said first code section.

4. (Previously Presented) The method of claim 1, wherein said first language is a
directive language, and said second language is a selected one of XML and an object-
oriented language.

5. (Previously Presented) The method of claim 1, wherein said first language is
an object-oriented language, and said second language is XML.

6. (Previously Presented) The method of claim 1, wherein the method further
comprises

recognizing a third code section with at least code statements of a third
programming language; and

invoking a third code statement processing unit of the third programming
language to process the third code section.

7. (Original) The method of claim 6, wherein said third code section is embedded within said second code section, and said second code section is embedded within said first code section.

8. (Previously Presented) The method of claim 6, wherein said first language is a directive language, said second language is an object-oriented language, and said third language is XML.

9. (Previously Presented) The method of claim 1, wherein the method further comprises

recognizing an invocation of a library function within at least a selected one of
said first and second code sections; and

invoking the library function, and outputting the result of the invocation.

10. (Original) The method of claim 9, wherein the library function is a selected one of an emit function for outputting execution results, a pop function for returning an element, and a push function for backing up an insertion point.

11. (Original) The method of claim 1, wherein the method further comprises

recognizing a header section of a selected one of the first and the second
programming language;

recognizing a directive statement within the header section, enumerating one or
more data packages; and

importing the enumerated one or more data packages for use within code
sections with at least statements of the selected first and second
programming language.

12. (Original) The method of claim 1, wherein the method further comprises

recognizing a header section of a selected one of the first and the second programming language;

recognizing a declare statement within the header section, enumerating one or more processing methods; and

instantiating the enumerated one or more processing methods for use within code sections with at least statements of the selected first and second programming language.

13. (Original) The method of claim 1, wherein the method further comprises

recognizing a header section of a selected one of the first and the second programming language;

recognizing a declare statement within the header section, enumerating one or more instance variables; and

instantiating the enumerated one or more instance variables for use within code sections with at least statements of the selected first and second programming language.

14.-19. (Cancelled)

20. (Previously Presented) An apparatus comprising:

at least one storage unit having stored thereon programming instructions designed to instantiate an execution engine to enable the apparatus to read, by the execution engine, a data processing representation having code sections with code statements of at least a first and a second programming language,

recognize, by the execution engine, a first code section with code statements of at least the first programming language,

invoking, by the execution engine, a first code statement processing unit of the first programming language to process the first code section,

recognize, by the execution engine, a second code section with code statements of at least the second programming language,

invoking, by the execution engine, a second code statement processing unit of the second programming language to process the second code section; and

at least one processor coupled to said at least one storage unit to execute said programming instructions.

21. (Previously Presented) The apparatus of claim 20, wherein the first and second code sections are non-interleaved code sections.

22. (Original) The apparatus of claim 20, wherein said second code section is embedded within said first code section.

23. (Previously Presented) The apparatus of claim 20, wherein said first language is a directive language, and said second language is a selected one of XML and an object-oriented language.

24. (Previously Presented) The apparatus of claim 20, wherein said first language is an object-oriented language, and said second language is XML.

25. (Previously Presented) The apparatus of claim 20, wherein the programming instructions further enable the apparatus to

recognize a third code section with at least code statements of a third programming language; and

invoke a third code statement processing unit of the third programming language to process the third code section.

26.   (Original) The apparatus of claim 25, wherein said third code section is embedded within said second code section, and said second code section is embedded within said first code section.

27.   (Previously Presented) The apparatus of claim 25, wherein said first language is a directive language, said second language is an object-oriented language and said third language is XML.

28.   (Previously Presented) The apparatus of claim 20, wherein said programming instructions further enable the apparatus to
   recognize an invocation of a library function of a selected one of the first and
      the second programming language within the first code section; and
   invoke the library function, and output the result of the invocation.

29.   (Original) The apparatus of claim 28, wherein the library function is a selected one of an emit function for outputting execution results, a pop function for returning an element, and a push function for backing up an insertion point.

30.   (Original) The apparatus of claim 20, wherein the said programming instructions are further designed to enable the apparatus to
   recognize a header section of a selected one of the first and the second
      programming language;
   recognize a directive statement within the header section, enumerating one or
      more data packages; and
   import the enumerated one or more data packages for use by code sections
      with at least code statements of the selected one of the first and the
      second programming language.

31.   (Original) The apparatus of claim 20, wherein said programming instructions are further designed to enable the apparatus to

recognize a header section of a selected one of the first and the second
    programming language;

recognize a declare statement within the header section, enumerating one or
    more processing methods; and

instantiate the enumerated one or more processing methods for use within code
    sections with at least code statements of the selected one of the first and
    the second programming language.

32.   (Original) The apparatus of claim 20, wherein said programming instructions
are further designed to enable the apparatus to

recognize a header section of a selected one of the first and the second
    programming language;

recognize a declare statement within the header section, enumerating one or
    more instance variables; and

instantiate the enumerated one or more instance variables for use code
    sections with at least code statements of the selected one of the first and
    the second programming language.

33.-38.   (Cancelled)

Appendix B – Copies of Evidence Submitted

No evidence has been submitted under 37 C.F.R. 1.130, 1.131, or 1.132. No evidence entered by Examiner has been relied upon by Appellants in the appeal.

Appendix C – Related Proceedings

There are no related appeals or interference proceedings currently pending, which would directly affect or be directly affected by or have a bearing on the Board's decision in this appeal.